Übungen zur Vorlesung: Wissensbasierte Systeme

Blatt 2

Exercise 2.1:
Consider the following knowledge base:

has_access(X, library) <- student(X).
has_access(X, library) <- faculty(X).
has_access(X, library) <- has_access(Y, library) ∧ parent(Y,X).
has_access(X, office) <- has_keys(X).
faculty(diane).
faculty(ming).
student(william).
student(mary).
parent(diane, karen).
parent(diane, robyn).
parent(susan, sarah).
parent(sarah, ariel).
parent(karen, mary).
parent(karen, todd).

a) Provide a top-down SLD derivation of the query ?has_access(todd, library).
b) The query ?has_access(mary, library) has two straightforward, but quite distinct SLD derivations. Give both of them.
c) Does there exist an SLD derivation for ?has_access(ariel, library)? Briefly, why or why not?
d) Argue that the set of answers to the query ?has_access(X, office) is empty. if the clause
   has_keys(X) <- faculty(X).
   is added to KB, what is the set of answers to this query?

Exercise 2.2:
Give a most general unifier of the following pairs of expressions:

a) p(f (X), g(g(b))) and p(Z, g(Y))
b) g(f (X), r(X),t) and g(W, r(Q), Q)
c) bar(val(X, bb), Z) and bar(P, P)

Exercise 2.3:
Consider the problem of finding a path in the grid shown in Figure 4.12 from the position s to the position g. A piece can move on the grid horizontally and vertically, one square at a time. No step may be made into a forbidden shaded area.

a) On the grid shown in the figure, number the nodes visited (in order) for a depth-first search from s to g, given that the order of the operators you will test is: up left, right, then down. Assume there is a cycle check. A node is visited when it is taken off the frontier.

b) For the same grid, number the nodes visited, in order, for a best-first search from s to g. Manhattan distance should be used as the evaluation function. The Manhattan distance between two points is the distance in the x-direction plus the distance in the y-direction. It corresponds to the distance traveled along city streets arranged in a grid. Assume that you have multiple-path pruning. What is the first path found?

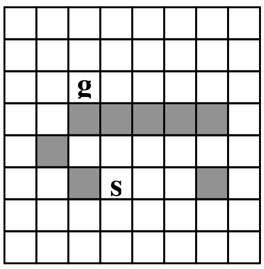
c) On the same grid, number the nodes visited, in order, for a heuristic depth-first search from s to g, given Manhattan distance as the evaluation function. Assume a cycle check. What's the path found?

d) Number the nodes in order for an A* search for the same graph. What's the path found?

e) Assume that you were to solve the same problem using dynamic programming. Give the dist value for each node, and show which path is found.

f) Based on this experience, discuss which algorithms are best suited for this problem.

g) Suppose that the graph extended infinetely in all directions. That is, there is no boundary, but s, g and the blocks are in the same relative positions to each other. Which methods would no longer find a path? Which would be the best method, and why?



Exercise 2.4:
Consider the following (real world) problem:

> Due to Bill's excessive drinking, it has been decided to limit him to exactly one liter of beer a day. You have only a 17-liter jug and a 7-liter jug. You can (1) fill either jug from a keg, (2) drink the contetns of a jug, or (3) pour beer from one jug to the other. Unfortunately, you are hopeless at guessing quantities. Given only these three operations, how can you restrict Bill's beer drinking to exactly one liter a day? (That is, how can you get one liter of beer in a jug if you only can perform the above three actions?)

The solution to this problem can be reduced to a graph-searching algorithm.

(a) Give the structure of a node in the graph.
(b) Define the neighbours of an arbitrary node.
(c) Draw the first two levels of a breadth-first search from the initial state of both jugs being empty.
(d) What is a suitable search strategy to use for this graph? Why?